Engineering Realistic Synthetic Insider Threat (Cyber-Social) Test Data

Kurt Wallnau
Senior Member of Technical Staff

Dr. Kurt Wallnau joined the SEI in 1993. He joined CERT Science of Cyber-Security (SoCS) in 2012 and is currently PI of the DARPA/ADAMS Red Team. Kurt has led and contributed to many SEI efforts to advance the theory and practice of software engineering. From 2010-12 he contributed to NSF's XSEDE program, defining its software and systems engineering process and from 2011-12 was manager of software development. Prior to that Kurt was PI of SEI's Predictable Assembly from Certifiable Components (PACC) initiative, which integrated software model checking, real-time analysis, and program generation to create software with predictable runtime behavior backed by verifiable evidence.

© 2013 Carnegie Mellon University

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
08 AUG 2013		**00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Engineering Realistic Synthetic Insider Threat (Cyber-Social) Test Data	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**15**	

My Topic Today

We need to protect ourselves from the “Insider Threat.”

- Part of solution: systems that *monitor and control* social behavior.

What engineering discipline is effective for assuring cyber-*social* systems?

- Those systems that produce value by exploiting the laws of human nature (as distinct from cyber-physical systems that exploit the laws of nature)
- But this is too broad a scope!

A narrower focus: How do we test systems whose dynamics are based in human nature that is (at best) partially understood?

- We are excluding from consideration simple “tripwire” systems
- Our concern is with the technology emerging from the intersection of
  - Big data machine learning analytics
  - Many forms of monitoring data

© 2013 Carnegie Mellon University

Key Takeaways

Cyber-social systems pose big challenges to systems engineering:

- Cyber-social: System Test → Human Subject Experiment?
- Human subject experiments are hard: we need synthetic social behavior

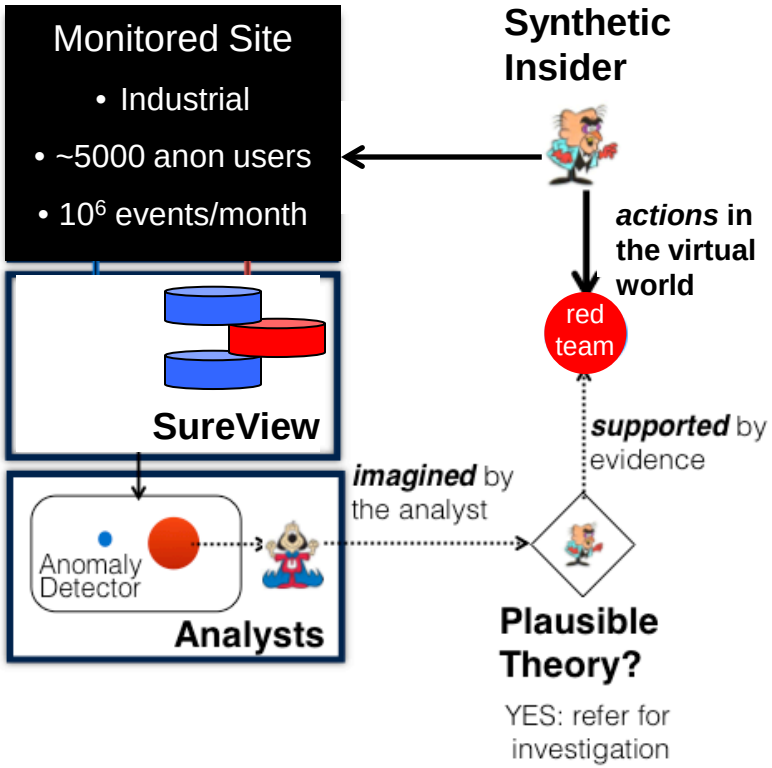
What is “real” in social behavior is great question for philosophers, but for engineers “realistic” is – and should be – a practical matter

- Realistic “enough” for the problem at hand
- Subject to the same “tradeoffs” as any other engineered artifact

Engineers make use of the sciences where possible but never wait for the sciences when it social needs dictate that solutions be built…

© 2013 Carnegie Mellon University

DARPA ADAMS – Insider Threat Detection



Examples will be drawn from experiences in DARPA/ADAMS[1]

- Anomaly Detection at Multiple Scales
- Connect The Dots technology
- Insider Threat demonstration domain
- Using host-based sensor data provided by an industry partner
  - Users are de-identified with strong protections on the use of data

CERT provides Red Team data

1. http://www.darpa.mil/Our_Work/I2O/Programs/Anomaly_Detection_at_Multiple_Scales_(ADAMS).aspx

© 2013 Carnegie Mellon University

Detecting Insiders: the “Haystack” Metaphor

Metaphor has tremendous power in the “cyber” world

- This “outsized” impact is acquired from nature of software itself
- Seasoned designers choose governing metaphors very carefully!

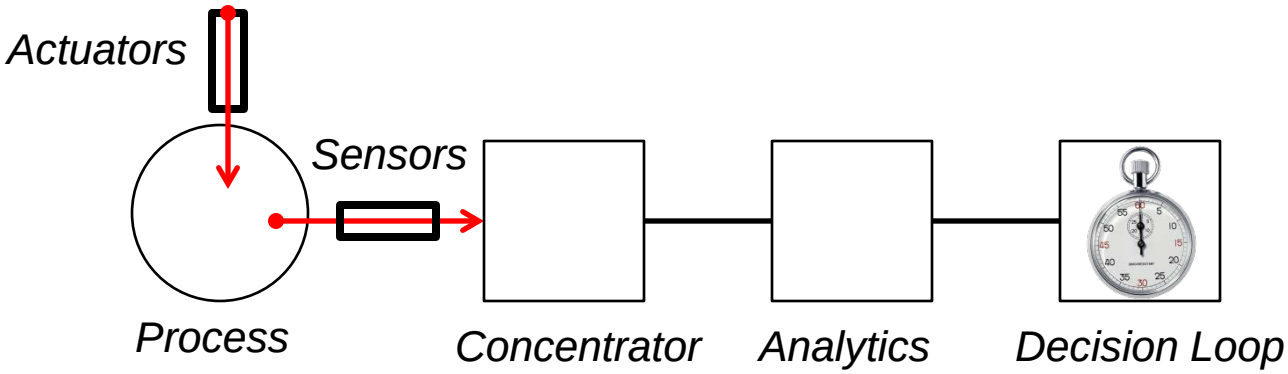
The Haystack metaphor is apt, descriptive, but not operational

- There is lots and lots of (human/social) data being collected
- Almost all of this data is innocuous (all but “the needle”)
- A tiny faction of this data is important (for some purpose)
- There are many haystack/silos, many needles to correlate

© 2013 Carnegie Mellon University

The “Control Systems” Metaphor

Control systems provide a better operational metaphor for testing:



However, social phenomena are real in a different way than physical ones

- They are real because we say they are: social reality is *constructed*[1]
- We’ve decided what is “real” by choosing what it is we “observe”

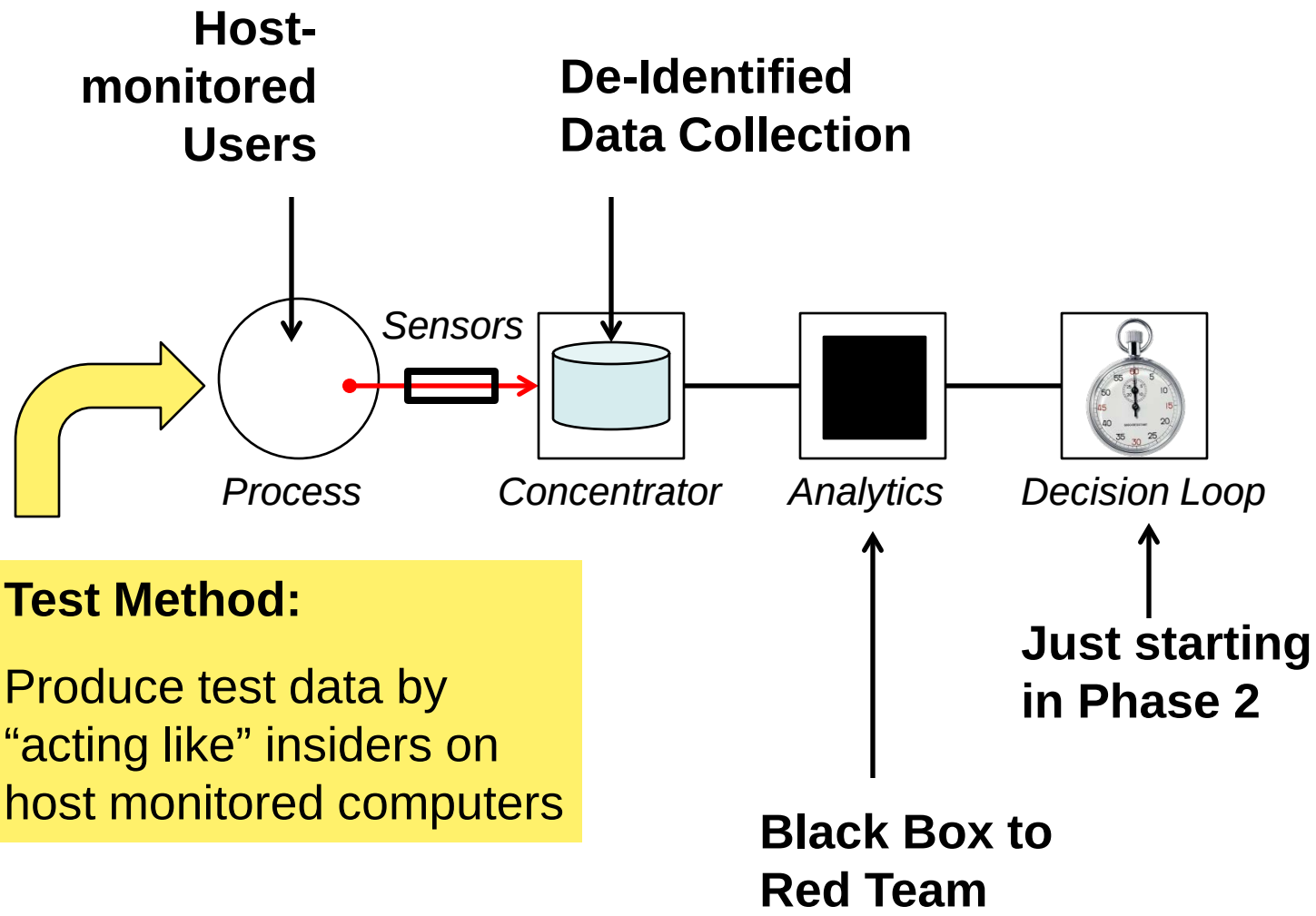
This is not circular – it is how humans create their social systems

- and why “realistic” must be defined with respect to context of use

For a non-technical but careful discussion see “The Construction of Social Reality,” John Searle, Free Press, 1995

© 2013 Carnegie Mellon University

ADAMS Insider Threat Detection (Gross Level)

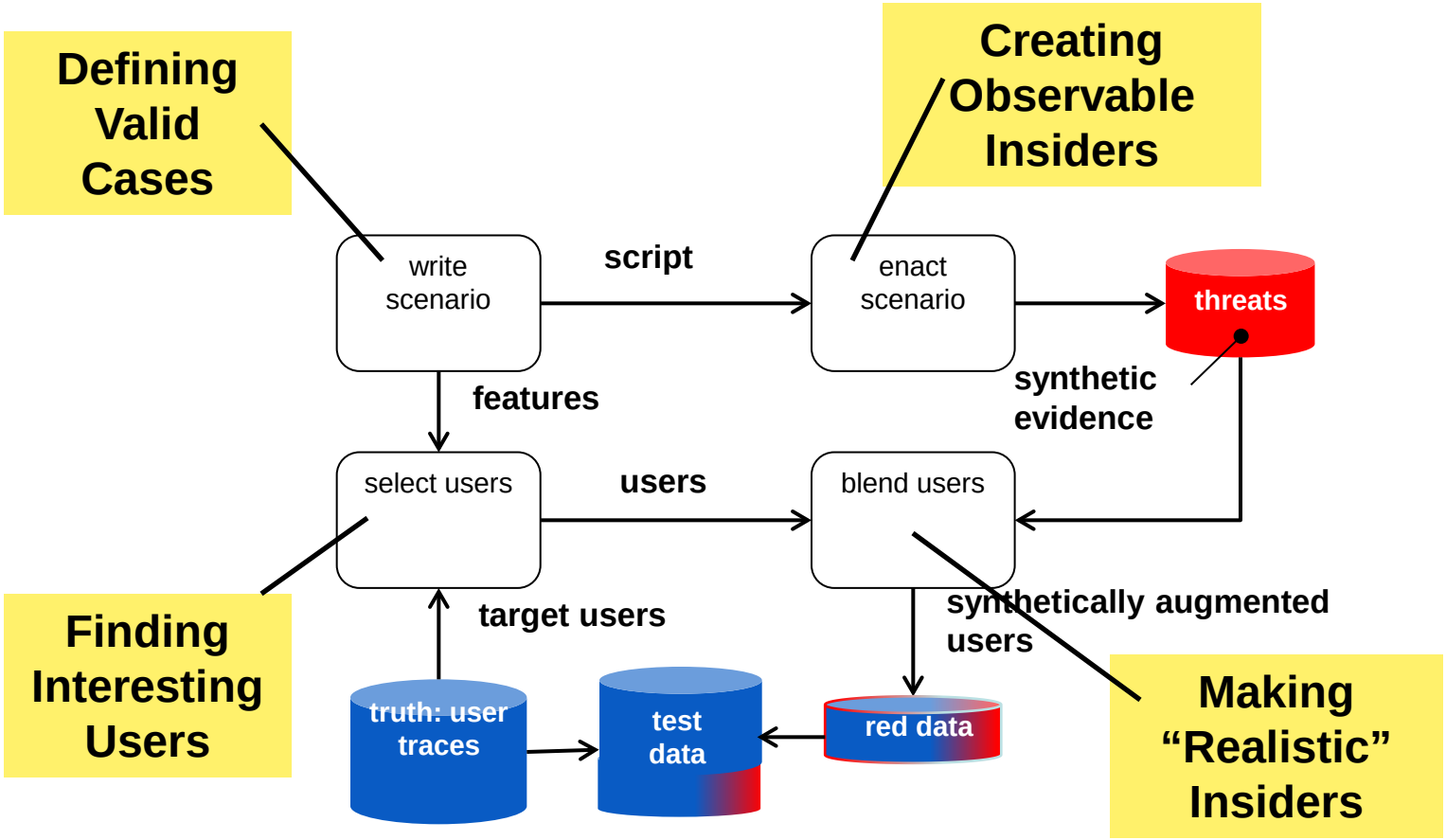


Test Method:

Produce test data by "acting like" insiders on host monitored computers

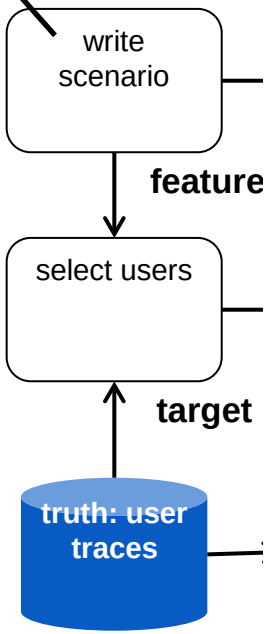
© 2013 Carnegie Mellon University

Process for Producing Insider Threat Data



© 2013 Carnegie Mellon University

Defining Valid Cases



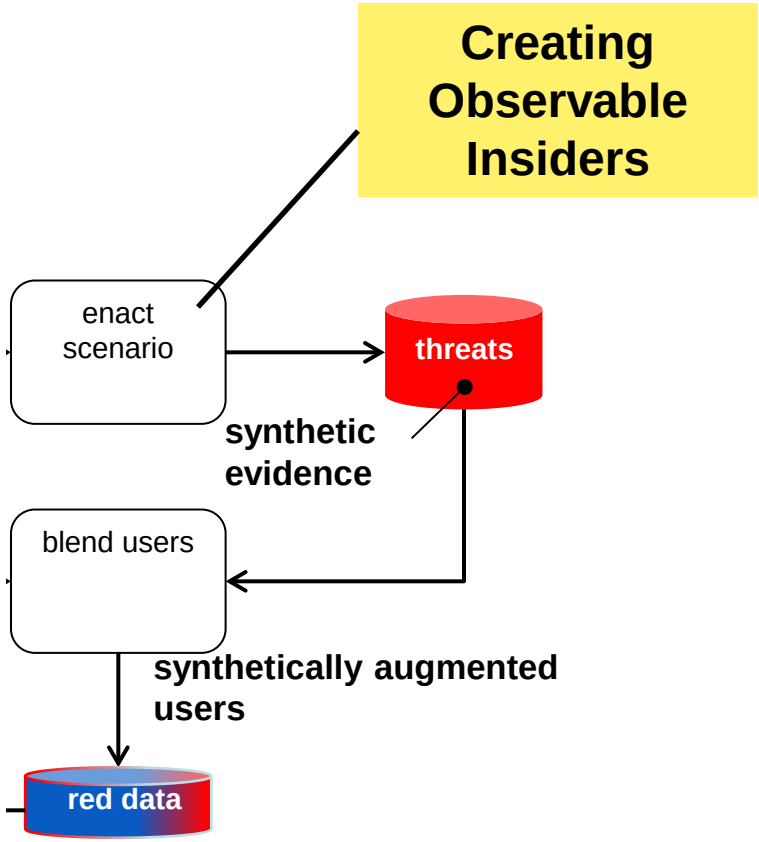
Validity is a kind of realism

- We assert there exists “insider behavior”
  - The insider threat community “constructs” this reality
- Validity is obtained by sampling these behaviors
  - Scenarios are a “judgment sampling” technique
- How do we validate the sample of a constructed reality?
  - That’s a hard question for science
  - It’s not a well-formed question for engineering

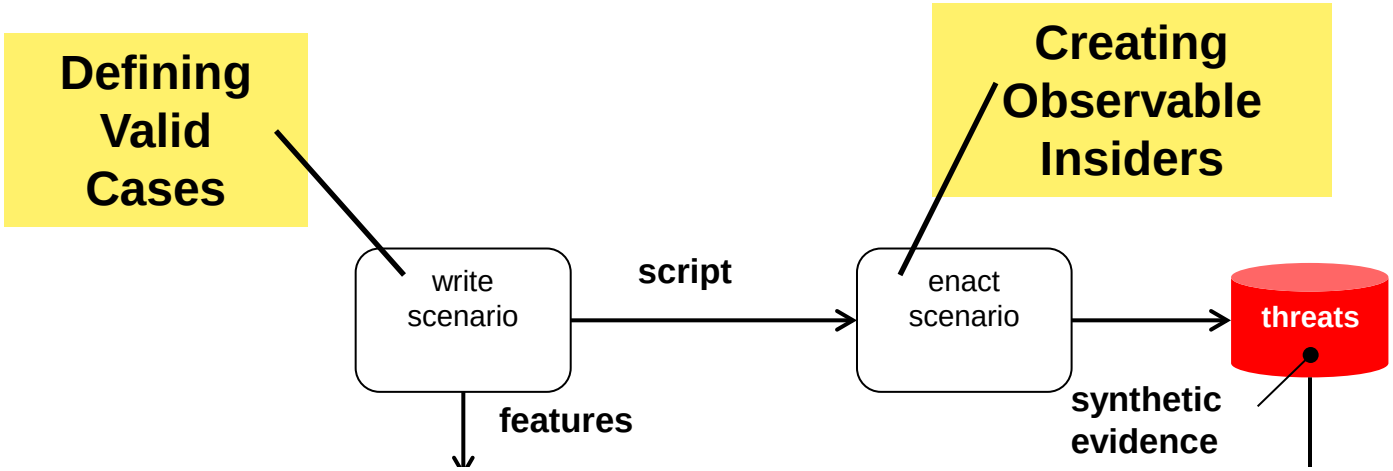
© 2013 Carnegie Mellon University

Observable Behavior and Sensors

- Are tests that produce no observable behavior useful?
- We can choose to make insider behavior more or less observable.
- We can choose different ways to make a behavior observable.
- In traditional testing we would expect as criteria, for example:
  - Sensor coverage
  - Signal strength per sensor
  - Code coverage (on analytics)



© 2013 Carnegie Mellon University



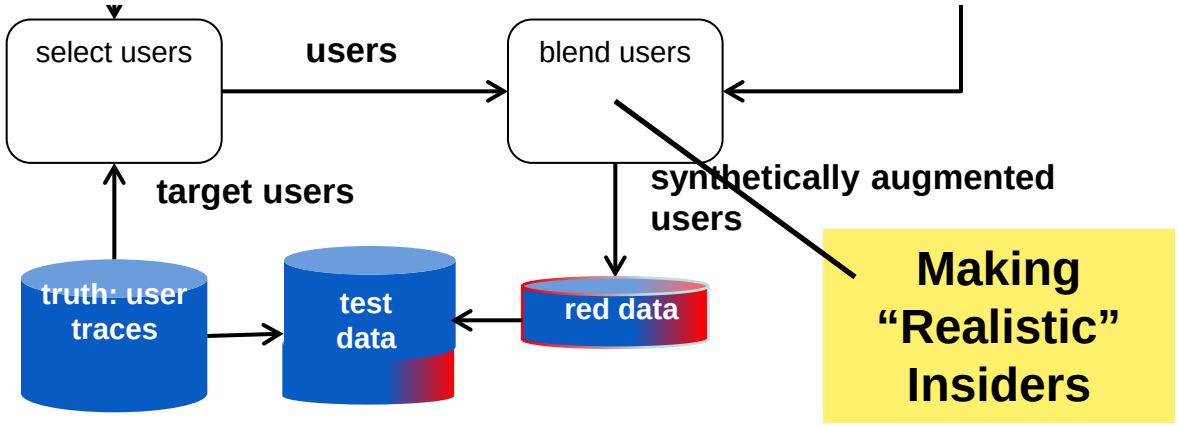
Validity and Observability: Science and Engineering Tradeoffs

- We are often confronted by this conundrum:
  - Would any real insider behave in the ways we require them to behave just so we can make their actions observable?
- Can a valid scenario be biased to ensure that it is observable?
  - The objective of test isn't to establish that an insider who knows the collection policy could *escape* detection
  - Endowing insiders with "realistic tradecraft" is itself an engineering concern in the way we design scenarios

© 2013 Carnegie Mellon University

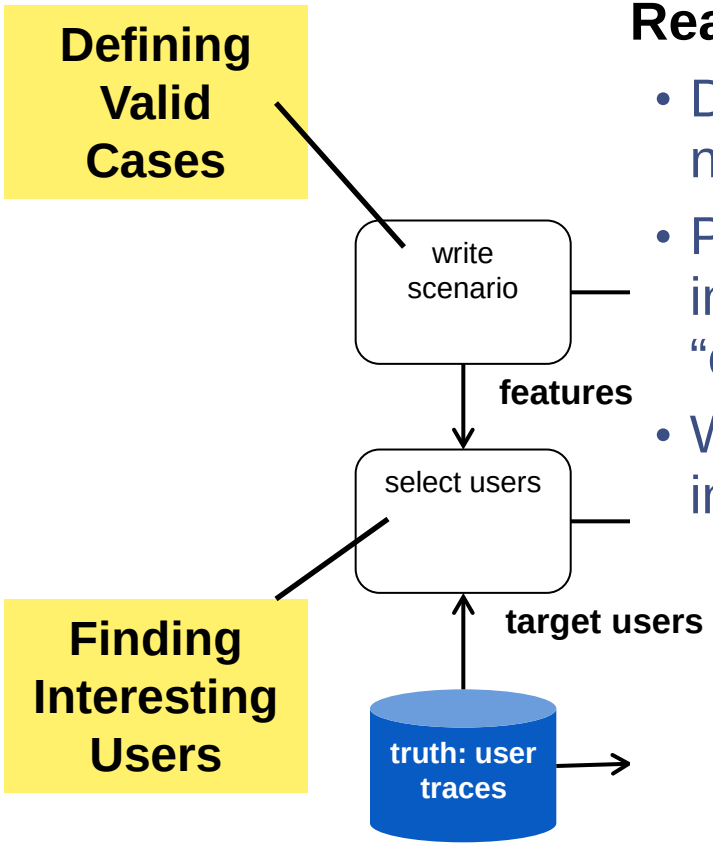
Realism and Artifacts

- The most concrete interpretation of realism is: can the synthetic data be distinguished from the real data it simulates?
- An indicator of synthetic origin is called an “artifact”
  - Intended : the “moral” of the scenario
  - Unintended: anything else



- Our technique of “augmenting” real users with synthetic behavior lets us “piggy-back” on real behavior and minimize the ratio of real-to-synthetic behavior in our data
  - But there are many subtle sources of artifact, e.g. email style

© 2013 Carnegie Mellon University



Realism and Validity Beyond Actions

- Detection is more than spotting late night USB (you all know that!)
- Personality traits, cognitive styles, interpersonal patterns…are the "context" for interpreting user actions.
- We select users for "blending" that are interesting in a variety of ways:
  - They typically do the things done by scenario characters
    - Realism – avoid artifacts
  - The do not typically do these things
    - Validity – change of behavior as indicator

© 2013 Carnegie Mellon University

Closing Thoughts

I have tried to persuade you that realism in social test data:

- Requires in operational context which establishes “how much” and “what kind” of realism is required
  - a decision procedure in an operational setting
  - engineering or engineering research purpose such as sensitivity testing
- Is a product of engineering tradeoff, usually made with an incomplete understanding of the social theories underlying the systems being tested.

It is not the case that “realism” is an intrinsic quality of the data

Programs such as ADAMS, and the technology we produced to construct test data, offers a way for the insider community to:

- Define scenarios narratives and characters with specific “traits”
- Specify how traits are mapped to (site-specific) data
- Generate test scenarios that can be relocated across different sites

© 2013 Carnegie Mellon University

Copyright 2013 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFCEA or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0000554